AGILENT TECHNOLOGIES, INC.
Legal Department, DL429
Intellectual Property Administration
P. O. Box 7599
Loveland, Colorado 80537-0599

ATTORNEY DOCKET NO. 10001846-01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Alan S Krech Jr, et al.

Serial No.: 09/659256 **Examiner: Shrader, Lawrence J.**

Filing Date: September 11, 2000 **Group Art Unit: 2124**

Title: Method And Apparatus For No-Latency Conditional Branching

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on December 9, 2004 .

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) **$500.00**.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)(1)-(5)) for the total number of months checked below:

☐	one month	$ 120.00
☐	two months	$ 450.00
☐	three months	$1020.00
☐	four months	$1590.00

☐ The extension fee has already been filled in this application.

☒ (b) Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **50-1078** the sum of $500.00 . At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account **50-1078** pursuant to 37 CFR 1.25.

A duplicate copy of this transmittal letter is enclosed.

Respectfully submitted,

Alan S Krech Jr, et al.

By June L Bouscaren

June L Bouscaren
Attorney/Agent for Applicant(s)

Reg. No. 37,928

Date: 02/08/05

Telephone No. 970 679 3238

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date of Deposit: 02/08/05 **OR**

☐ I hereby certify that this paper is being facsimile transmitted to the Patent and Trademark Office on the date shown below.

Date of Facsimile:

Typed Name: Shirley A. Gonzales

Signature: Shirley A. Gonzales

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Matter of the Application of: Krech et al.
Serial No. : 09/659,256
Filed: September 11, 2000
For : Method and Apparatus For No-latency Conditional Branching
Examiner: Shrader
Group Art Unit: 2124

Appeal Brief

Real Party in Interest

The real party in interest is Assignee Agilent Technologies, Inc. All inventors were employees of Agilent at the time of conception and reduction to practice of the claimed invention.

Related Appeals and Interferences

There are no known related appeals and interferences.

Status of Claims

Claims 1-18 are pending and are rejected.

Status of Amendments

No amendments are filed subsequent to the last final rejection.

Summary of the Invention

Claim 1:

An apparatus provides for conditional branching in an IC test program without inserting latency in the test sequence as it processes the branching condition. A sequencer 19 executes program instructions and one or more of the program instructions is a conditional branch instruction. See p. 18, lines 9-12. The conditional branch instruction specifies a branch condition address 300. See p. 19, lines 18-22. Whether to branch or not is based upon appropriate selection of one of available flags 25, 55 that indicate a status of the specified condition. See p. 19, line 26-29. A programmable flag selection memory 502 (shown as 32 different registers) is programmed prior to test execution and permits a programmer to define up to 32 different conditions that may be used for purposes of conditional branching in any one program. See p. 22, lines 18-19 and p. 23, lines

20-27. Each register in the programmable flag selection memory 502 selects one of the available flags 25,55 presented to respective first flag selectors 504. See p. 22, lines 5-7. An output of each first flag selector 504 provides a selected flag 508 for presentation to the second flag selector 506. See p. 22, lines 14-19. A branch condition address 300 selects which one 304 of the flags 508 is presented at an output of the second flag selector 506 as the branch flag 304 for the current conditional instruction. See p. 22, lines 16-18.

Claim 2:

Branch flags 304 from multiple branch units 302 are logically combined to arrive at a branching bit 306. See FIG. 3.

Claim 12:

A method compiles source code that contains one or more conditional branching instructions. The steps comprise identifying each conditional branch instruction and for each condition represented in the conditional branch instruction, determining one or more flags 25, 55 upon which the condition is

based. See p. 25, line 22 through p.26, line 5. The compiler generates values that are stored into the programmable flag selection memory register elements 502. See FIGs 7 & 5 and p. 26, lines 10-21. The compiler then assigns a value for the branch condition address 300 and encodes the branch condition address value in the conditional branch instruction. See p. 26, lines 22-29. Values for the flag selection memory register elements and instructions for the conditional branching are stored with program object code. See p. 27, lines 2-6.

Claim 13:

The compiler identifies each conditional branching instruction and reorders the flags to a set placement format. See. P. 26, lines 5-10. Because only a limited number of conditions are used per program before reloading the flag selection memory, the re-ordering permits the compiler to make more efficient use of the memory by recognizing redundant use of conditions without requiring the programmer to follow more rigorous programming formats.

Claim 14:

The compiler applies DeMorgan's theorem to the condition designation to convert disjunctive operations to the conjunctive equivalent in order to properly format for use of the conjunctive hardware operation. See p. 25, lines 22 through p. 26, line 4 and FIG. 3.

Claim 15:

An apparatus provides a compiler, sequencer 19 and branch unit 302 to perform conditional branching in an IC tester without inserting latency in the test sequence as it processes the branching condition. The compiler converts test program source code having one or more conditional branch instructions into object code executable by the sequencer 19. The compiler assigns values for a branch condition address 300 and flag selection memory 502. See FIG 5 and p. 23, lines 20-21 and p. 25, lines 18-21. The sequencer 19 executes the object code where one or more instructions is a conditional branch instruction specifying the branch condition address 300. The apparatus further comprises a branch unit 302 (FIG 3) that includes a

programmable flag selection memory 502 and first flag selectors 504. See FIG 5. Each first flag selector 504 accepts available flags 25, 55 and selects one of them based upon the contents of the flag selection memory 502. A second flag selector 506 accepts the selected flags 508 and selects one of them as a branch flag 304 based upon the branch condition address 300. The branch flag 304 indicates to the sequencer 19 whether or not to branch.

Grounds of Rejection

Claims 1-11:

Claims 1, 3-5, and 7 are rejected as anticipated under 35 U.S.C. §102(b) by US Pat. No. 5,991,868 to Kamiyama (herein "the Kamiyama patent").

Claims 2 and 6 are rejected as rendered obvious under 35 U.S.C. §103 by the Kamiyama patent in view of US Pat. No. 5,991,868 to Asakawa (herein "the Asakawa patent").

Claim 8 is rejected as rendered obvious under 35 U.S.C. §103 by the Kamiyama patent in view of US Pat. No. 5,274,770 to Khim Yeoh et al.(herein "the Khim patent").

Claims 9-11 are rejected as rendered obvious

under 35 U.S.C. §103 by the Kamiyama patent in view of the Asakawa patent and further in view of the Khim patent.

Claims 12-14:

Claims 12-13 are rejected as rendered obvious under 35 U.S.C. §103 by U.S. Pat. No. 6,546,550 to Ogata (herein "the Ogata patent") in view of the Kamiyama patent.

Claim 14 is rejected as rendered obvious by the Ogata patent in view of US Pat. No. 5,276,776 to Grady et al. (herein "the Grady patent").

Claims 15-18

Claims 15 & 18 are rejected as rendered obvious under 35 U.S.C. §103 by the Kamiyama patent in view of US Pat. No. 4,742,466 to Ochiai (herein "the Ochiai patent").

Claim 16 is rejected as rendered obvious under 35 U.S.C. §103 by the Kamiyama patent in view of the Ochiai patent and further in view of the Grady patent.

Claim 17 is rejected as rendered obvious under 35 U.S.C. §103 by the Kamiyama patent in view of the Ochiai patent and further in view of the Grady

patent and further in view of US Pat. No. 6,272,599 to Prasanna.

Argument

Claims 1-11:

Applicant respectfully suggests that the Kamiyama patent does not present a prima facie case of anticipation under 35 U.S.C. §102. The rejection of claim 1 as stated in Office Action dated September 10, 2004, maintains that the Kamiyama patent discloses a branch unit (FIG 6 of the Kamiyama patent) comprising a flag selection memory 106 programmed with a plurality of values. See clause 4 of the 9/10/2004 Office Action. The rejection further maintains that the Kamiyama patent discloses a first flag selector (branch judging unit 107) and a second flag selector (condition judging unit 7) selecting one as a branch flag based upon a branch condition address. Ibid. A proper anticipation rejection requires that each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. MPEP §2131 citing Verdegaal Bros. v. Union Oil Co. of California, 814 F. 2d. 628 (Fed. Cir. 1987). The Office Action states that "Kamiyama

discloses...a branch unit (FIG 6, item 107) that selects flags read out of a memory". Applicant understands the position taken in clause 14 of the 9/10/2004 Office Action to be that the program status word 106 disclosed in the Kamiyama patent is the same as Applicant's recited element of "a flag selection memory, capable of being programmed with a plurality of selection values, each selection value providing independent selection input into a respective plurality of first flag selectors". In reply, Applicant agrees that the program status word 106 disclosed in the Kamiyama patent is a memory structure. Multiplexers, however, accept two different types of input. A first type of input to a multiplexer is the data input. A second type of input to the multiplexer is the selection input that determines which one of the data input is to be presented at an output of the multiplexer. The PSW 106 disclosed in the Kamiyama patent provides *data input*. See FIG. 6 in the Kamiyama patent. The flag selection memory 502 as claimed provides *selection input* and the available flags 25, 55 provide the data input to the first flag selectors 504. See claim 1 and FIG 5 of the present Specification. There is no express or implied suggestion in the

Kamiyama patent to modify the teachings so that a memory structure provides the selection input as suggested in the Office Action. Accordingly, the prima facie case of anticipation is not presented. Additionally, Applicant respectfully suggests that the statement that claim 1 recites "a branch unit …that selects flags read out of a memory, then that flag is used to determine a branch condition" is an incorrect paraphrasing of the term "memory" as it is used in claim 1. See Office Action dated 9/10/2004, clause 4. Applicants flag selection memory 502 provides the selection input for first flag selectors 504 for selecting among some number of flag selector inputs (i.e. the available flags 25, 55). The available flags 25,55 are disclosed, but not claimed, to be available from ALUs and not from a memory structure. See element 25 in FIGs 4&5. A source of the available flags 25, 55 is not an element of claim 1. Applicant understands after careful reading of the Kamiyama patent teachings, that the PSW 106 is the source of the flags used for a branching indication. See Col. 4, lines 39-42 of the Kamiyama patent. The flag selection memory 502 of claim 1 is not the source of the available flags, but is the selection input 502 of the first flag

selector 504 to select one of the data input or available flags 25,55. See FIG. 5. Claim 1 recites a "flag selection memory… programmed with… selection values" where "each… value provides independent selection input into a respective plurality of first flag selectors". As a first point, the Kamiyama patent does not teach that the PSW 106 is programmed with values, but stores data that is sourced from the operation of a calculator 104. See col. 4, lines 7-10. The flags from the calculator as disclosed in the Kamiyama patent are more similar to the available flags 25, 55 in Applicant's disclosure that are disclosed, but not claimed to come from the ALUs rather than the flag selection memory 502 as claimed. As a second point, the Kamiyama patent does not teach that the value in the flag selection memory (PSW 106) is a value that determines which flag of the available flags is used. See p. 22, lines 5-7 of the Specification where it is taught "the selection value loaded into the flag selection memory selects one flag from a set of available flags" and claim 1 language "each…first flag selector presenting at an output a…selected flag from a plurality of available flags". By contrast Kamiyama teaches that the PSW 106 contains the flags

upon which the branch/no branch decision is based. See col. 4, lines 13-17 and FIGURE 6. As a third point, the Kamiyama patent does not teach "independent selection input" because input to multiplexing elements 6-1 through 6-4 is a value designating whether flags from an 8-bit or 16-bit flag group *as a whole* are presented to the condition judging unit 7. The determination (via a flag group designation signal) is sourced from the instruction decoding unit 103 and Applicant does not find any further disclosure regarding the actual source of the flag designation signal. See FIG 5 and col. 4, lines 43-47 of the Kamiyama patent. Because the flags in the Kamiyama patent are selected as a group via the flag group designation signal, the selection is not "independent selection" as claimed, and because the selection is apparently made as part of the instruction decoding process, it is not stored in a flag selection memory as claimed. In any case, it is apparent from FIGURE 6 in the Kamiyama patent that the element that the 9/10/2004 Office Action indicates is the flag selection memory (i.e. PSW 106) does not provide "selection input into... the first flag selectors, each...first flag selector presenting at an output a ...first stage selection

flag from a plurality of available flags based upon each said selection value" as recited in claim 1. An illustration of the indirect nature of the flag selection according to the present teachings is illustrated in FIGURE 5 of the present patent application, where flag selection registers 502 are programmed with values that determine which flag from a plurality of available flags 25, 55 is presented to a second flag selector 506. By contrast, only direct flag selection is taught in the Kamiyama patent after designation via a flag group designation signal of an 8-bit or 16-bit flag group. See FIG. 6 and elements 3-1 through 3-4 and 4-1 through 4-4 of the Kamiyama patent. Because all elements and limitations of the claimed invention are not disclosed in the Kamiyama patent, anticipation is not shown. Withdrawal of the anticipation rejection of claim 1 is respectfully requested.

Claims 2-11:

Claims 2-11 are also rejected and incorporate the rejection of claim 1. Claims 2-11 depend from claim 1 or an intervening dependent claim. Accordingly, the rejection of claims 2-11 is

believed to be improper for at least the same reasons as the rejection of claim 1 is believed to be improper and withdrawal of the rejections of claims 2-11 is respectfully requested.

With respect to claim 2, the Asakawa patent teaches execution of conditional branch instructions in a pipelined process where condition codes are settled by condition code setting instructions at different stages of the pipelined process. See col. 2, lines 50-56 of the Asakawa patent. Claim 2 depends from claim 1 and, therefore, includes all of the elements and limitations of claim 1 in addition to the elements and limitations claimed in claim 2. Obviousness requires that 1) the combination of references must teach or suggest all elements and limitations, 2) there must be motivation to modify or combine the teachings of the cited art, and 3) there must be a reasonable expectation of success. MPEP §2142. The combination of the Kamiyama and Asakawa patents does not disclose all elements and limitations of claim 2. Neither the Kamiyama nor the Asakawa patents teach or suggest to one of ordinary skill a "plurality of branch units and … an operator that accepts …the branch flags and logically combines the branch flags to create a

branching bit indicating whether the sequencer is to branch". Neither the Kamiyama not the Asakawa patent teach or suggest the elements and limitations comprising the branch unit 302 of claim 1. Claim 2 recites multiple branch units 302 and an operator 316 that accepts the branch flags 304 to generate a branching bit 306. See FIG 3. The Asakawa patent teaches pipelined stages with intermediate settling of condition codes. See col. 5, lines 49-66 of the Asakawa patent. The AND gate operator shown in FIG 7 of the Asakawa patent indicates to the circuit at what stage a condition code is settled. See col. 5, line 49 through col. 6, line 10. By contrast, the operator 316 as claimed in claim 2 "accepts a plurality of the branch flags 304", which are determined in parallel in the different branch units 302 and "logically combines the branch flags to create a branching bit 306". Pipelining and condition code settling is not implicated in the claimed invention as all branch flags 304 are determined in parallel for the same conditional instruction. See FIG. 3. Accodingly, a prima facie case of obviousness is not established for claim 2

Claim 12:

The position taken in the 9/10/2004 Office Action is that art that teaches dynamic compiler execution, the Ogata patent, in combination with the teachings of the Kamiyama patent renders claim 12 obvious because teachings with respect to a compiler combined with the Kamiyama teachings would suggest the compiler of claim 12 to one of ordinary skill in the art. A proper obviousness rejection requires all elements and limitations of the rejected claim be present or suggested in the combination of the prior art together with a suggestion or motivation to combine to arrive at the claimed invention. MPEP §2142. Applicant respectfully suggests that the combination of the Ogata and Kamiyama patents does not establish a prima facie case of obviousness. The Ogata patent is cited to establish that compilers for conditional branch instructions are known. Applicant does not dispute this fact. The Ogata patent, however, does not include any teachings related to the relationship between a flag selection value and flag selection array elements and assigning a branch condition address and an output of the compiler. Accordingly, in order for the obviousness rejection to stand, all elements and

limitations of claim 12 must be found in the Kamiyama patent and then be an inherent part of a compiler operation. Neither the Kamiyama patent nor the Ogata patent teaches use of a flag selection value. Neither the Kamiyama patent nor the Ogata patent teaches "identifying a flag selection value... and storing... said flag selection value in a respective one of two or more flag selection register array elements" as claimed. Specifically, the Kamiyama patent teachings that the PSW 106 is a result of calculator operations and does not suggest that the contents of the flag selection memory are determined by a compiler prior to test execution. See col. 4, lines 13-18. Neither the Kamiyama patent nor the Ogata patent teaches use of "two or more flag selection register array elements" as claimed. Because some of the elements and limitations found in claim 12 are not disclosed, inherent or suggested in the cited prior art, it is not possible for the combination of the two to render obvious the compiler of claim 12 without resorting to impermissible hindsight reconstruction. Accordingly,, there is no basis for a prima facie case of obviousness and withdrawal of the obviousness rejection of claim 12 is respectfully

requested.

Claims 13-14:

Claims 13 and 14 are also rejected as rendered obvious by the Kamiyama and Ogata patents. The reasons for the rejection incorporate the reasons for rejection of claim 12. Claims 13 and 14 depend from claim 12. Accordingly, the rejection of claims 13 and 14 is believed to be improper for at least the same reasons as the rejection of claim 12 is believed to be improper and withdrawal of the rejections of claims 13 and 14 is respectfully requested.

With respect to claim 13, the "re-ordering" step of claim 13 is not a calculator operation as suggested in claim 10 of the 9/10/2004 Office Action, but a compiler operation to inject programming discipline to permit a compiler to recognize reuse of conditions to take advantage of programming redundancies without requiring a programmer to practice the discipline. See p. 26, lines 4-10 of the present Specification. The teachings in Kamiyama with respect to the calculator are in the context of test program execution and not compiler execution. There are no teachings or

suggestion in either the Kamiyama or the Ogata patent regarding use of multiple flags to determine a branch condition. Accordingly, there can be no teachings in either patent regarding reordering the multiple flags in the compiler operation to comply with a set flag placement format. Therefore, the combination of the Kamiyama and Ogata patents cannot provide a prima facie case of obviousness of claim 13. Withdrawal of the rejection of claim 13 is respectfully requested.

With respect to claim 14, there are no teachings in either the Kamiyama nor the Ogata patent with respect to application of DeMorgan's theorem on conditional branch conditions. Applicant does not dispute that DeMorgan's theorem is known. Applicant maintains, however, that an additional feature that is made possible by the compiler invention of claim 12 from which claim 14 depends providing efficiencies gained through application of DeMorgan's theorem that permits programmer use of conjunctive and disjunctive logical combinations of flags as programmed using the flag selection memory is not taught or suggested by Kamiyama and Ogata patents. Accordingly, the combination of Kamiyama and Ogata does not provide a prima facie case of

obviousness of claim 14 and withdrawal of the rejection of claim 14 is respectfully requested.

Claims 15-18:

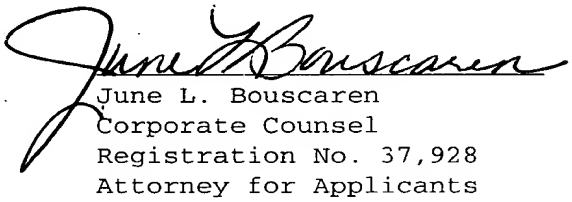
The Ochiai patent discloses a compiler for conditional branch instructions and teaches setting a branch/through flag. The 9/10/2004 Office Action suggests that it would have been obvious to one of ordinary skill in the art to combine the compiler as taught in the Ochiai patent with the teachings of the Kamiyama patent to arrive at the invention of claim 15. Applicant respectfully suggests that the combination of the Kamiyama and Ochiai patents does not present a prima facie case of obviousness. Obviousness requires that 1) the combination of references must teach or suggest all elements and limitations, 2) there must be motivation to modify or combine the teachings of the cited art, and 3) there must be a reasonable expectation of success. MPEP §2142. Neither the Kamiyama patent nor the Ochiai patent teaches a flag selection memory. Therefore, neither the Kamiyama patent nor the Ochiai patent nor the combination teaches or suggests "a compiler assigning… values for a flag selection memory" as claimed. The PSW 106 disclosed

in the Kamiyama patent is a memory structure, but values stored in the PSW 106 are calculated during program execution in calculator 104. See col. 4, lines 13-21 of the Kamiyama patent. One of ordinary skill in the art equates compiler operations with program calculations only it the most general sense of both being processor operations. Therefore, one of ordinary skill in the art would not be led to assign values to the PSW 106 in a compiler operation because the PSW 106 is the result of a calculator operation. Neither the Kamiyama patent nor the Ochiai patent teaches a "flag selection memory capable of being programmed with a plurality of selection values, each selection value providing independent selection input into a respective plurality of first flag selectors" as claimed. The PSW 106 taught in the Kamiyama patent provides data input but does not provide "selection input into first flag selectors" as claimed. The flag group designation signal appears to provide the selection input for the circuit of FIG 6 of the Kamiyama patent, but no teachings indicate that the compiler determines the value of the flag group designation signal. Therefore, one of ordinary skill in the art would not be led to create a compiler that assigns

selection values for the flag selection memory as claimed. Because elements of the claim are not found in either of the cited references, it is not possible for the combination of the cited references to render obvious the apparatus comprising a compiler as claimed. Accordingly, a prima facie case of obviousness is not established and withdrawal of the obviousness rejection is respectfully requested.

Respectfully submitted,

Krech et al.
Applicant(s)



June L. Bouscaren
Corporate Counsel
Registration No. 37,928
Attorney for Applicants

Agilent Technologies, Inc.
Legal Dept. DL 432
815 14th St. S.W.
Loveland, CO 80537
Phone: (970) 679-3238
Fax: (970) 679-5772

Appendix

Current status of Pending Claims

1. An apparatus for conditional branching comprising:

a sequencer executing a plurality of program instructions, one or more of said program instructions including a conditional branch instruction, said conditional branch instruction specifying a branch condition address and a conditional instruction,

a branch unit comprising a flag selection memory capable of being programmed with a plurality of selection values, each selection value providing independent selection input into a respective plurality of first flag selectors, each respective first flag selector presenting at an output a respective first stage selected flag from a plurality of available flags based upon each said selection value, a second flag selector accepting a plurality of said first stage selected flags and selecting one of said first stage selected flags to present as a branch flag based upon said branch condition address, said branch flag indicating to said sequencer whether to branch according to said conditional instruction.

2. An apparatus as recited in claim 1 and further comprising a plurality of said branch units and further comprising an operator that accepts a respective plurality of said branch flags and logically combines said branch flags to create a branching bit, said branching bit indicating whether said sequencer is to branch according to said conditional instruction.

3. An apparatus as recited in claim 1, said flag selection memory comprising a plurality of programmable registers.

4. An apparatus as recited in claim 1, each said first flag selectors comprising a multiple input, single output multiplexer.

5. An apparatus as recited in claim 1, said second flag selector comprising a multiple input single out multiplexer.

6. An apparatus as recited in claim 2, said operator comprising a multiple input logical AND operator.

7. An apparatus as recited in claim 1, wherein said branch address comprises a plurality of bits in said conditional branch instruction.

8. An apparatus as recited in claim 1 and further comprising a b1not0 bit in said conditional branch instruction directing said sequencer whether it is to branch on a one or a zero of said branch flag.

9. An apparatus as recited in claim 2 and further comprising a b1not0 bit in said conditional branch instruction directing said sequencer whether it is to branch on a one or a zero of said branch flag.

10. An apparatus as recited in claim 9 and further comprising a dual input selector accepting said branching bit and an inverse of said branching bit, said b1not0 bit operating on said dual input selector.

11. An apparatus as recited in claim 10, wherein said dual input selector is a dual input single output multiplexer.

12. A method for compiling source code containing one or more conditional branching instructions comprising the steps of:

interpreting the source code, the source code comprising a plurality of program instructions,

identifying each conditional branch instruction in said source code, and for each conditional branch~~ing~~ instruction, determining a set of flags as a subset of all available flags_upon which each said conditional branch instruction is based, identifying a flag selection value for each flag in said set of flags, and storing each said flag selection value in a respective one of two or more flag selection register array elements, assigning a branch condition address for said conditional branching instruction, encoding said branch condition address in a binary representation of said conditional branching instruction, and storing said encoded one or more conditional branching instructions and said flag selection register array elements in an object code format.

13. A method for compiling source code as recited in claim 12, the step of identifying each conditional branching instruction further comprising the step of re-ordering said set of flags to a set placement format.

14. A method for compiling source code as recited in claim 12 and further comprising the step of converting all disjunctive operations to a conjunctive equivalent.

15. An apparatus for conditional branching comprising:

a compiler for converting source code including one or more conditional branch instructions into object code, the compiler assigning values for a branch condition address and values for a flag selection memory,

a sequencer executing said object code comprising one or more of said conditional branch instructions, each said conditional branch instruction specifying a branch condition address and a conditional instruction, and

a branch unit comprising said flag selection memory capable of being programmed with a plurality of selection values, each selection value providing independent selection input into a respective plurality of first flag selectors, each respective first flag selector presenting at an output a respective first stage selected flag from a plurality of available flags based upon each said selection value, a second flag selector accepting a plurality of said first stage selected flags and selecting one of said first stage selected flags to present as a branch flag based upon said branch condition address, said branch flag indicating to said sequencer whether to branch according to said

conditional instruction.

16. An apparatus for conditional branching as recited in claim 15, said compiler also converting disjunctive logical operations specified in each said conditional branch instructions to an equivalent conjunctive logical operation.

17. An apparatus for conditional branching as recited in claim 16, said compiler setting a b1not0 bit for said program instruction if said logical operation is converted from said disjunctive logical operation to said equivalent conjunctive logical operation.

18. An apparatus for conditional branching as recited in claim 15, and further comprising one or more arithmetic logic units that supply said plurality of available flags.